Esame di Linguaggi di Programmazione – CdL Informatica
Appello del 9 giugno 2020 - Soluzione

Quesito 1 (PUNTI 2)

Quale delle seguenti affermazioni è corretta?

a. **La classe dei linguaggi accettati da automi a stati finiti deterministici è contenuta nella classe dei linguaggi liberi da contesto.**

b. La classe dei linguaggi accettati da automi a stati finiti deterministici è equivalente alla classe dei linguaggi liberi da contesto.

c. La classe dei linguaggi accettati da automi a stati finiti deterministici è equivalente alla classe dei linguaggi monotoni.

d. La classe dei linguaggi accettati da automi a stati finiti deterministici è equivalente alla classe dei linguaggi dipendenti da contesto.

Quesito 2 (PUNTI 2)

Quale delle seguenti affermazioni è corretta?

a. La classe dei linguaggi liberi da contesto è contenuta nella classe dei linguaggi di tipo 3.

b. La classe dei linguaggi liberi da contesto è contenuta nella classe dei linguaggi accettati da automi a stati finiti deterministici.

c. La classe dei linguaggi liberi da contesto contiene la classe dei linguaggi regolari.

d. La classe dei linguaggi liberi da contesto contiene la classe dei linguaggi di tipo 0.

Quesito 3 (PUNTI 2)

Quale delle seguenti affermazioni è corretta?

a. La classe dei linguaggi dipendenti da contesto è equivalente alla classe dei linguaggi di tipo 3.

b. La classe dei linguaggi dipendenti da contesto è equivalente alla classe dei linguaggi accettati da automi a stati finiti deterministici.

c. La classe dei linguaggi dipendenti da contesto è contenuta nella classe dei linguaggi di tipo 0.

d. La classe dei linguaggi dipendenti da contesto è equivalente alla classe dei linguaggi regolari.

Quesito 4 (PUNTI 2)

Quale delle seguenti affermazioni è corretta?

a. **La classe dei linguaggi regolari è chiusa rispetto all'operazione di complemento.**

b. La classe dei linguaggi liberi da contesto è chiusa rispetto all'operazione di complemento.

c. La classe dei linguaggi di tipo 0 è chiusa rispetto all'operazione di complemento.

d. La classe dei linguaggi monotoni non è chiusa all'operazione di complemento.

Quesito 5 (PUNTI 2)

Quale delle seguenti affermazioni è corretta?

a. **La classe dei linguaggi dipendenti da contesto è chiusa rispetto alle operazioni di unione, concatenazione, iterazione, complemento, intersezione.**
b. La classe dei linguaggi liberi da contesto è chiusa rispetto alle operazioni di unione, concatenazione, iterazione, complemento, intersezione.
c. La classe dei linguaggi regolari non è chiusa rispetto alle operazioni di unione, concatenazione, iterazione, complemento, intersezione.
d. La classe dei linguaggi di tipo 0 è chiusa rispetto alle operazioni di unione, concatenazione, iterazione, complemento, intersezione.

Quesito 6 (PUNTI 3)

Sia data la seguente grammatica libera da contesto G=(X, V, S, P) con
X = {a, b, c}, V = {S, S1, S2, S3} e
P = {S -> S1S2S3, S1 -> ab|aS1b, S2 -> b|bS2, S3 -> bc|bS3c}
Quale delle seguenti affermazioni è corretta?

a. $L(G) = \{w \in X^* \mid w = a^n b^m c^k, m \geq n+k, n > 0, k > 0\}$
b. $\boldsymbol{L(G) = \{w \in X^* \mid w = a^n b^m c^k, m > n+k, n > 0, k > 0\}}$
c. $L(G) = \{w \in X^* \mid w = a^n b^m c^k, m \leq n+k, n > 0, k > 0\}$
d. $L(G) = \{w \in X^* \mid w = a^n b^m c^k, m < n+k, n > 0, k > 0\}$

Quesito 7 (PUNTI 5)

Descrivere le operazioni di *set* e *reset* per la gestione delle tabelle dei simboli nei linguaggi a blocchi.

Per i linguaggi a blocchi sono necessarie due operazioni addizionali che chiamiamo set e reset. Set si invoca quando si entra in un blocco, reset quando si esce. Questo è necessario perché in un linguaggio a blocchi una variabile con lo stesso nome può essere dichiarata in punti diversi del programma ed assumere attributi diversi, e quindi si deve assicurare che per ogni istanza di un nome di variabile sia associato un unico entry nella tabella dei simboli. Quindi ad ogni inizio di un blocco l'operazione di set attribuisce una nuova sotto tabella per gli attributi delle variabili dichiarate nel nuovo blocco. Supponiamo che le sottotavole siano create attribuendole numeri interi crescenti. Se la ricerca comincia nella sottotavola N e poi procede fino alla sottotavola 1, la variabile che si trova è l'ultima che si è inserita (secondo le regole di scope di linguaggi a blocchi) ed è eliminata ogni ambiguità. All'uscita del blocco, l'operazione di reset rimuove l'ultima sottotabella allocata. Le variabili di quel blocco, infatti, non possono più essere referenziate. Le tabelle dei simboli per i linguaggi a blocchi devono dunque essere organizzate a pila (stack).

Quesito 8 (PUNTI 6)

Sia dato il seguente automa riconoscitore a stati finiti *non deterministico*:

$$M = (Q, \delta, q_0, F)$$

con alfabeto di ingresso $X = \{a, b\}$, ove:

$$Q = \{q_0, q_1, q_2\}$$

$\delta(q_0, a) = \{q_1\}$ $\quad$ $\delta(q_0, b) = -$
$\delta(q_1, a) = \{q_2\}$ $\quad$ $\delta(q_1, b) = -$
$\delta(q_2, a) = \{q_1, q_2\}$ $\quad$ $\delta(q_2, b) = \{q_2\}$

ed $F = \{q_2\}$.

Costruire un automa a stati finiti *deterministico* *M'* equivalente ad *M*.

$M' = (Q', \delta', q'_0, F')$

$Q' = 2^Q$

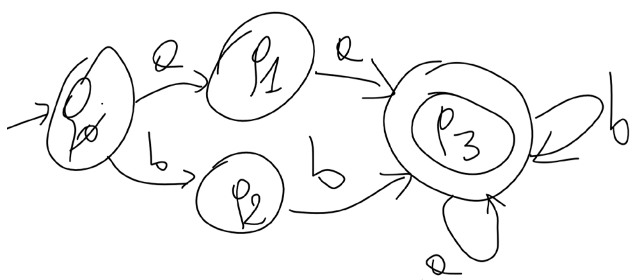
$q'_0 = \{q_0\}$

δ': $Q' \times X \rightarrow Q'$
$\delta'(\{q_0\}, a) = \{q_1\}$ $\quad$ $\delta'(\{q_0\}, b) = \emptyset$
$\delta'(\{q_1\}, a) = \{q_2\}$ $\quad$ $\delta'(\{q_1\}, b) = \emptyset$
$\delta'(\{q_2\}, a) = \{q_1, q_2\}$ $\quad$ $\delta'(\{q_2\}, b) = \{q_2\}$
$\delta'(\{q_1, q_2\}, a) = \{q_1, q_2\}$ $\quad$ $\delta'(\{q_1, q_2\}, b) = \{q_2\}$

$F' = \{p \subseteq Q \mid p \cap F \neq \emptyset\} = \{\{q_2\}, \{q_1, q_2\}\}$

Quesito 9 (PUNTI 4)

Sia data il seguente automa a stati finiti M con alfabeto di ingresso X = {a, b}:



Quale delle seguenti affermazioni è corretta?

a. $\boldsymbol{T(M) = \{w \in X^* \mid w = aa\beta, \beta \in X^*\} \cup \{w \in X^* \mid w = bb\beta, \beta \in X^*\}}$

b. $T(M) = \{w \in X^* \mid w = aa\beta, \beta \in X^*\} \cap \{w \in X^* \mid w = bb\beta, \beta \in X^*\}$

c. $T(M) = \{w \in X^* \mid w = aa\beta,\ \beta \in X^*\} \bullet \{w \in X^* \mid w = bb\beta,\ \beta \in X^*\}$

d. $T(M) = \{w \in X^* \mid w = ba\beta,\ \beta \in X^*\} \bullet \{w \in X^* \mid w = ab\beta,\ \beta \in X^*\}$

Quesito 10 (PUNTI 5)

Dimostrare che la classe dei linguaggi liberi da contesto è inclusa strettamente nella classe dei linguaggi context-sensitive.

Osserviamo che le produzioni C.F. sono un caso particolare delle produzioni C.S. con l'unica eccezione rappresentata dalle produzioni:

$$A \to \lambda, \qquad A \neq S$$

che sono C.F. ma ***non*** C.S.

Dunque:

$$\forall L : L \text{ linguaggio C.F.} \overset{def}{\Leftrightarrow} \exists G,\ G \text{ è C.F.} : L = L(G).$$

Se $A \to \lambda,\ A \in V \setminus \{S\}$ non è una produzione di *G*, allora *G* è anche C.S. (di tipo '1') e l'asserto è dimostrato. Il problema sorge se *G* ha almeno una λ-produzione. In tal caso, ci avvaliamo del Lemma della stringa vuota.

<u>Lemma della stringa vuota</u>

Sia *G* = (*X*, *V*, *S*, *P*) una grammatica C.F. con almeno una λ-produzione. Allora esiste una grammatica C.F. G' tale che:

i) $L(G) = L(G')$ (G' è equivalente a *G*);

ii) se $\lambda \notin L(G)$ allora in G' non esistono produzioni del tipo $A \to \lambda$;

iii) se $\lambda \in L(G)$ allora in G' esiste un'unica produzione $S' \to \lambda$, ove S' è il simbolo iniziale di G' ed S' non compare nella parte destra di alcuna produzione di G'.

Se *G* ha almeno una λ-produzione, utilizziamo il Lemma della stringa vuota per determinare una grammatica C.F. G' equivalente a *G*, ma priva di λ-produzioni (al più, in G' compare la produzione $S' \to \lambda$, ed S' non compare nella parte destra di alcuna produzione di G'). G' è di tipo '1'. Questo dimostra che *la classe dei linguaggi C.F. è inclusa nella classe dei linguaggi C.S.*

Dimostriamo ora che *la classe dei linguaggi C.F. non coincide con la classe dei linguaggi C.S.*

Il linguaggio $L = \{a^n b^n c^n \mid n > 0\}$ è nella classe dei linguaggi C.S., ma non nella classe dei linguaggi C.F.

Per dimostrarlo individuiamo innanzitutto una grammatica corretta per *L*:

G = (*X*, *V*, *S*, *P*),

$$X = \{a, b, c\} \qquad V = \{S, A, B, C\} \qquad P = \left\{ \begin{array}{l} S \overset{(1)}{\rightarrow} aBC,\ S \overset{(2)}{\rightarrow} aSBC \\ CB \overset{(3)}{\rightarrow} BC, \\ aB \overset{(4)}{\rightarrow} ab,\ bB \overset{(5)}{\rightarrow} bb, \\ bC \overset{(6)}{\rightarrow} bc,\ cC \overset{(7)}{\rightarrow} cc \end{array} \right\}$$

G è una grammatica monotona. Per il teorema di equivalenza delle classi dei linguaggi monotoni e contestuali (Teorema 3.2), esiste una grammatica contestuale G' equivalente a *G* corretta per *L*, dunque risulta dimostrato che *L* è nella classe dei linguaggi C.S.

Dimostriamo ora che *L* non è non nella classe dei linguaggi C.F.

Supponiamo, per assurdo, che *L* sia un linguaggio C.F. Per il Pumping Lemma sui linguaggi C.F., esiste una costante *p* tale che:

$$\forall z \ \ z \in L,\ |z| > p \ \ \Rightarrow \ \ z = uvwxy$$

ed inoltre valgono le seguenti:

(1) $|vwx| \le p$;

(2) $vx \ne \lambda$;

(3) $\forall i,\ i \ge 0 : uv^i wx^i y \in L$

Consideriamo una parola in *L*:

$$z = a^p b^p c^p \ .$$

Il Pumping Lemma può essere applicato a tale parola poiché $|z| = 3p > p$ e dunque *z* può essere scritta nella forma *z* = *uvwxy*, in modo tale che:

$$|vwx| \le p$$

Poiché la stringa *vwx* ha lunghezza al più uguale a *p*, si hanno le seguenti possibilità:

(i) *vwx* è formata da sole *a*, cioè è del tipo $vwx = a^k$, $0 < k \le p$;

(ii) *vwx* è formata da sole *b* , cioè è del tipo $vwx = b^k$, $0 < k \le p$;

(iii) *vwx* è formata da sole *c* , cioè è del tipo $vwx = c^k$, $0 < k \le p$;

(iv) *vwx* è a cavallo tra ***a*** e *b* , cioè è del tipo $vwx = a^k b^r$, $0 < k + r \le p$ e $k, r > 0$;

(v) *vwx* è a cavallo tra *b* e *c* , cioè è del tipo $vwx = b^k c^r$, $0 < k + r \le p$ e $k, r > 0$.

È immediato osservare che *vwx* non può essere formata da *a*, *b* e *c* (ossia non può essere contemporaneamente a cavallo tra *a* e *b* e tra *b* e *c*), perché non è sufficientemente lunga.

Consideriamo la stringa (*pompata*):

$$uv^2wx^2y$$

per ognuno dei casi (i) - (v).

Per la (3) del Pumping Lemma sui linguaggi liberi, si dovrebbe avere:

$$uv^2wx^2y \in L$$

Ma nel caso:

(i) aggiungiamo almeno una *a*, ed al più *p a*; $uv^2wx^2y = a^{p+t}b^pc^p$, $0 < t \leq p$;

(ii) aggiungiamo almeno una *b*, ed al più *p b*; $uv^2wx^2y = a^pb^{p+t}c^p$, $0 < t \leq p$;

(iii) aggiungiamo almeno una *c*, ed al più *p c*; $uv^2wx^2y = a^pb^pc^{p+t}$, $0 < t \leq p$;

(iv) per la (2) del Pumping Lemma, si hanno le seguenti possibilità:

(iv.a) $v \neq \lambda,\ x \neq \lambda$;
(iv.b) $v \neq \lambda,\ x = \lambda$;
(iv.c) $v = \lambda,\ x \neq \lambda$.

Osserviamo preliminarmente che, se $v \neq \lambda$, allora *v* è costituita da sole *a*. Infatti, se *v* fosse del tipo $v = a^kb^{r'}$, con $0 < r' \leq r$, si avrebbe $uv^2wx^2y = a^{p-k}a^kb^{r'}a^kb^{r'}b^sc^p \notin L$, con $p - r' \leq s \leq 2(r - r') + p - r$.

Analogamente, se $x \neq \lambda$, allora *x* è costituita da sole *b*. Infatti, se *x* fosse del tipo $x = a^{k'}b^r$, con $0 < k' \leq k$, si avrebbe $uv^2wx^2y = a^sa^{k'}b^ra^{k'}b^rb^{p-r}c^p \notin L$, con $p - k' \leq s \leq 2(k - k') + p - k$.

Per cui:

(iv.a) se $v \neq \lambda,\ x \neq \lambda$, per l'osservazione precedente $v = a^{k'}$, con $0 < k' \leq k$ e $x = b^{r'}$, $0 < r' \leq r$ e si ha che $uv^2wx^2y = a^{p+k'}b^{p+r'}c^p \notin L$, poiché $k',\ r' > 0$

(iv.b) se $v \neq \lambda,\ x = \lambda$, si ha $v = a^{k'}$, con $0 < k' \leq k$ e $uv^2wx^2y = a^{p+k'}b^pc^p \notin L$, poiché $k' > 0$

(iv.c) se $v = \lambda,\ x \neq \lambda$, si ha $x = b^{r'}$, con $0 < r' \leq r$ e $uv^2wx^2y = a^pb^{p+r'}c^p \notin L$, poiché $r' > 0$

(v) Si lascia per esercizio.

In ciascuno dei casi (i) - (v) $uv^2wx^2y \notin L$.

Assurdo. Ne segue che *L* non è un linguaggio C.F., poiché è un linguaggio per il quale non vale il Pumping Lemma per i linguaggi C.F.